ERICSSON CHALLENGE

Challenger: Carlos Rodriguez

Description of task

Selection of data

Connection MySQL – python connector and automation through python script

Connection MySQL- tableau connector and creation of dashboard for data filtering and visualization

Presentation of the work done

Selection data, connection and script automation

- Selection of data. https://github.com/dsindy/kaggle-titanic/blob/master/data/test.csv. Screenshots of the script of automation made with python are shown below. It is written verifying existing data in database or building it again once it is run, so there is no need to comment anything.

```
from mysql.connector import Error
import pandas as pd
import connection_details


def import_csv():
    titanicData = pd.read_csv('tested.csv')
    values = {"Age": 0, "Cabin": 'None', "Fare":0}
    titanicDataClean = titanicData.fillna(value=values)
    #print(titanicData.head())
    return titanicDataClean


def connection():
    try:
        global conn
        conn = msql.connect(
            host=connection_details.HOST,
            user=connection_details.USER,
            password=connection_details.PASSWORD,
            auth_plugin='mysql_native_password'
        )

        if conn.is_connected():
            print("Connection established")
    except Error as e:
        print("Error while connecting to MySQL", e)

def create_database():
    cursor = conn.cursor()
```

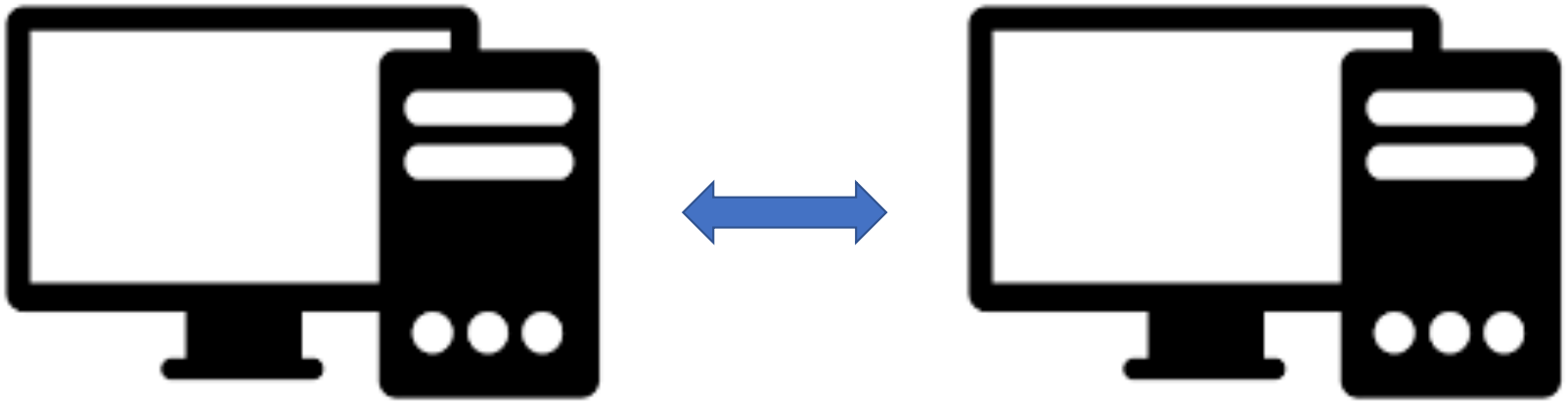
```
def create_table(database):
    cursor = conn.cursor()
    cursor.execute("USE " + database + ";")
    cursor.execute('DROP TABLE IF EXISTS passengers;')
    cursor.execute("CREATE TABLE passengers (PassengerId SMALLINT NOT NULL, Survived BOOLEAN NOT NULL,\
Pclass TINYINT, Name VARCHAR(100) NOT NULL,Sex CHAR(50), Age FLOAT(5,2), SibSp TINYINT, Parch TINYINT,\
Ticket VARCHAR(50),Fare FLOAT(6,2), Cabin VARCHAR(50), Embarked CHAR(1), PRIMARY KEY (PassengerId))")
    print("table passengers created.")
    for i,row in import_csv().iterrows():
        #print(row)
        sql_statement = "INSERT INTO titanic.passengers VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"
        cursor.execute(sql_statement,tuple(row))
        print("Record inserted "+ str(i))
        conn.commit()


def create_new_column_cities(database):
    cursor = conn.cursor()
    cursor.execute("USE " + database + ";")
    # create a procedure to drop column. if exists drop column
    cursor.callproc('schema_change')
    query_1 = "ALTER TABLE passengers ADD COLUMN cities CHAR(20) Default 'None';"
    cursor.execute(query_1)
    #query_2 ="UPDATE passengers SET cities = 'Cobh' WHERE embarked='Q';"
    query_2 = "UPDATE passengers SET cities = CASE WHEN embarked ='Q' THEN 'Cobh'\n\
WHEN embarked ='S' THEN 'Southampton'\n\
WHEN embarked='C' THEN 'Cherbourg'\n\
else cities\n\
end;"
    cursor.execute(query_2)
    conn.commit()


def close():
    conn.close()
```

Installation of Tableau, connection with MySQL

- Installation of Tableau in a new VM with Virtual Box with Windows 10 as OS. Bridge connection so they can talk to each other.



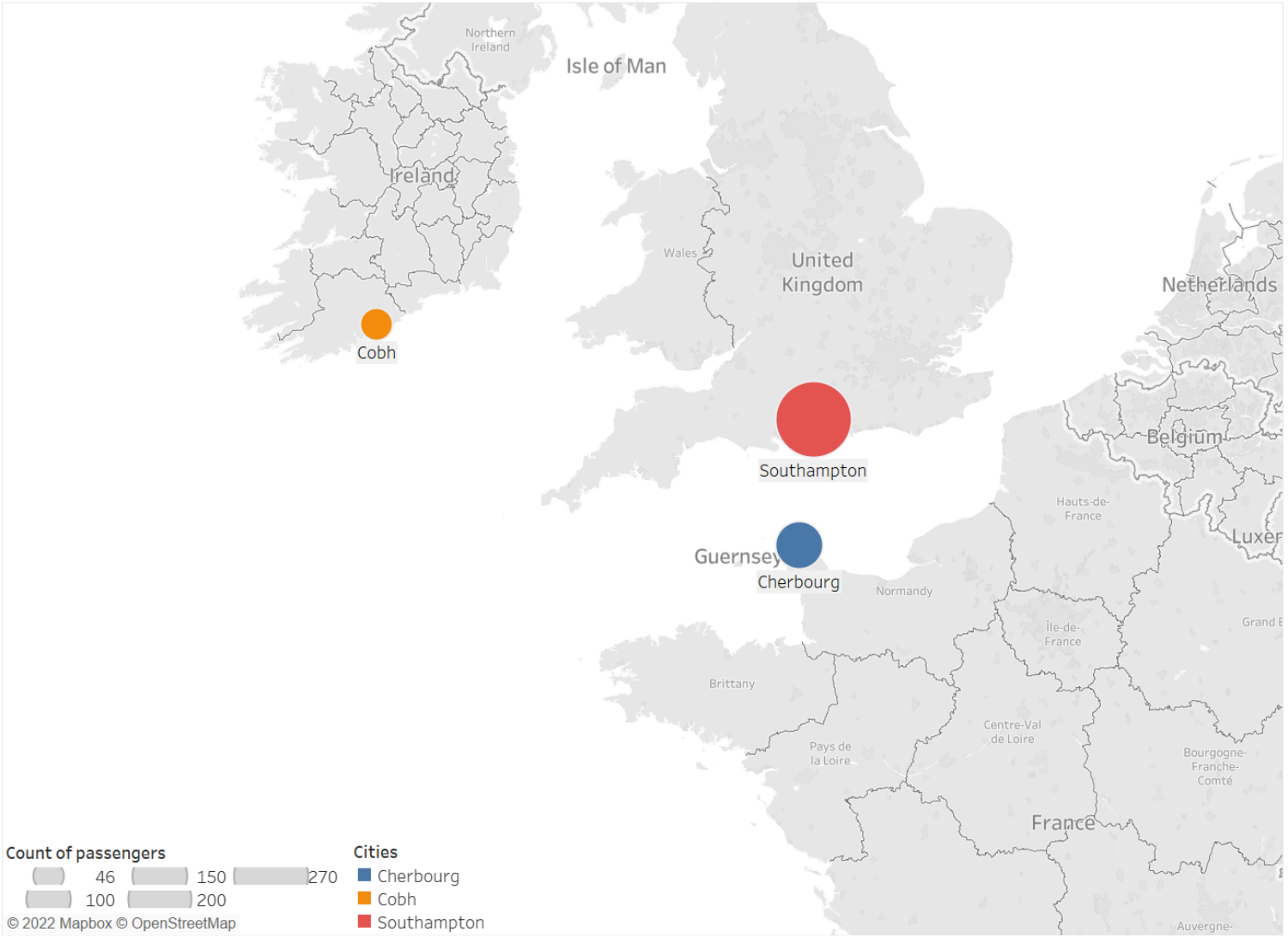
HOST OPERATING SYSTEM. MAC OS

Creation of new connection in MySQL 0.0.0.0:3306 and user with privileges to access the database from any network. (%)

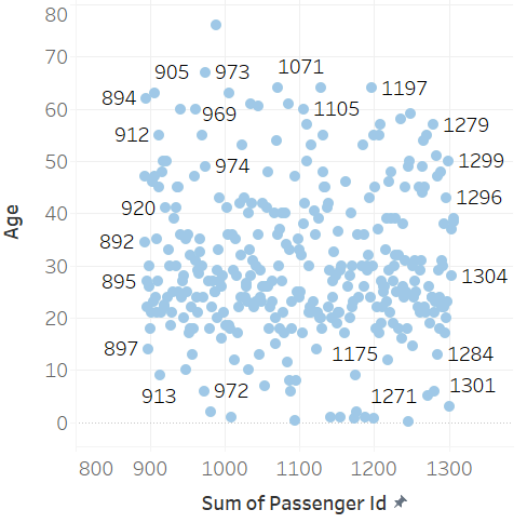
GUEST OPERATING SYSTEM. WINDOWS 10

Installation of MySQL connector for Tableau and insertion of the connection parameters and user authentication details.

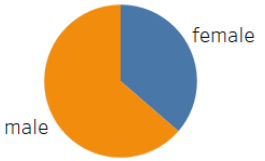
Departure cities by number of passengers.



Distribution of passengers by age



Male/female number of passengers



Sex
female
male

Challenges

- 1st challenge: valid connection with database from the script
- 2nd challenge: figuring out to avoid commenting out code in the script. For that creating a procedure to drop column. Practice with queries.
- 3rd challenge: installation of Tableau and connection with MySQL database.
- 4th challenge: creation of map with locations of the cities. For that adding a new column and identifying specifically the cities in Tableau.

Extra task, not demanded.

- Utilization of Git version control system. The code has been uploaded to Github, **https://github.com/carrod85/ericsson** and the code used and detailed explanation of the steps that were taken in Markdown format (file info.md)